

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	50	collaborative adj behavior	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/07 11:27
S2	36	framework adj semantic	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 14:35
S3	3	valid adj nesting	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/07 11:28
S4	1	S1 and S2	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/07 11:28
S5	51	semantic adj framework	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/07 11:29
S6	36	framework adj semantic	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 14:36
S7	29	S6 and (@pd<"20031231" or @ad<"20031231" or @prad<"20031231" or @rlad<"20031231")	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 07:23
S8	0	S7 and ((unit adj testing) or (unit adj test) or (JUnit) or (CUnit))	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 14:38
S9	6860	(unit adj testing) or (unit adj test) or (JUnit) or (CUnit)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 14:38
S11	89	JUnit	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 14:39

EAST Search History

S12	76	S11 and (@pd<"20031231" or @ad<"20031231" or @prad<"20031231" or @rlad<"20031231")	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 14:45
S16	3	S10 and (test adj case adj class)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:57
S17	437	S10 and (test adj case)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:41
S18	5968	S10 and (not camera)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:18
S20	4727	S18 and (not optical) and (not ink)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:20
S21	2879	S20 and (not wire) and (not nozzle) and (not key)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:22
S22	1087	S21 and (not telephone) and (not clock) and (not eyes) and (not visual) and (not ticketing) and (not circuit)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:26
S23	640	S22 and (not allergy) and (not tires) and (not dosage) and (not organic) and (not fuel) and (not esters) and (not drug) and (not elastomeric) and (not shampoo) and (not organometal) and (not blood) and (not vehicle) and (not membranes)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:32
S27	377	S23 and (not needle) and (not oxygen) and (not multicell) and (not earthquake) and (not temperature) and (not microfiltration) and (not cardboard) and (not bank) and (not pressure) and (not gas) and (not plastic) and (not elastic) and (not microorganisms) and (not greasy) and (not water) and (not soil) and (not enzyme) and (not liquid) and (not oil)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:38

EAST Search History

S29	363	S27 and (not dust) and (not torsion) and (not brake) and (not gear) and (not magnet) and (not trolley)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:41
S30	20	S29 and (test adj case)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:41
S31	354	S10 and frame\$1work	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:59
S32	109	S31 and (test adj case)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/11 18:59
S33	2111	(unit adj testing) and (@pd<"20031231" or @ad<"20031231" or @prad<"20031231" or @rlad<"20031231")	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 07:23
S34	316	S33 and (test adj case)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 07:24
S35	177	S33 and frame\$1work	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 07:24
S36	132	(frame\$1work adj testing)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 17:03
S37	122	S36 and (@pd<"20031231" or @ad<"20031231" or @prad<"20031231" or @rlad<"20031231")	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 17:04
S38	62	S37 and (not genes) and (not human) and (not cancer)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:40
S39	57	S38 and (not tire) and (not torsion)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/12 17:05

EAST Search History

S40	36	framework adj semantic	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:36
S41	89	JUnit	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:37
S42	89	JUnit	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:44
S43	1	("20020116153").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO	OR	OFF	2006/09/13 13:46
S44	808	(717/124).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO	OR	OFF	2006/09/13 13:50
S45	136	S44 and framework	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:50
S46	109	S45 and (@pd<"20031231" or @ad<"20031231" or @prad<"20031231" or @rlad<"20031231")	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:51
S47	14	S46 and ((unit adj test) or (unit adj testing) or (JUnit))	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/13 13:51
S57	1	("6601018").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO	OR	OFF	2006/09/14 08:09
S58	136	framework adj testing	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/14 08:12
S59	126	S58 and (@pd<"20031231" or @ad<"20031231" or @prad<"20031231" or @rlad<"20031231")	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/14 08:13

EAST Search History

S60	66	S59 and (not gene) and (not disease) and (not human)	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	ON	2006/09/14 08:13
-----	----	--	---	----	----	------------------

 [Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: [The ACM Digital Library](#) [The Guide](#)

JUnit

[Full Text](#) [PDF](#) [HTML](#) [BIB](#) [EndNote](#) [RefWorks](#)

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

and Published before December 2002
Term used **JUnit**

Found 17 of 17

Sort results by [relevance](#) [Save results to a Binder](#)
 [Search Tips](#) [Open results in a new window](#)

Display results [expanded form](#)

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 17 of 17

Relevance scale 

1 [Korat: automated testing based on Java predicates](#)

 Chandrasekhar Boyapati, Sarfraz Khurshid, Darko Marinov
July 2002 **ACM SIGSOFT Software Engineering Notes**, Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis ISSTA '02, Volume 27 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(171.43 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper presents Korat, a novel framework for automated testing of Java programs. Given a formal specification for a method, Korat uses the method precondition to automatically generate all (nonisomorphic) test cases up to a given small size. Korat then executes the method on each test case, and uses the method postcondition as a test oracle to check the correctness of each output. To generate test cases for a method, Korat constructs a Java predicate (i.e., a method that returns a boolean) fr ...

2 [Unit testing frameworks](#)

 Robert E. Noonan, Richard H. Prosl
February 2002 **ACM SIGCSE Bulletin**, Proceedings of the 33rd SIGCSE technical symposium on Computer science education SIGCSE '02, Volume 34 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(406.45 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

In this paper we explore frameworks for performing unit testing in Java. The vehicle for this exploration is a student-written, skeleton program developed for the *Computer Graphics* course. Our analysis of this one experiment leads us to speculate what benefits in program development and design might accrue by requiring students to unit test their own programs.

3 [Object oriented framework development](#)

 Marcus Eduardo Markiewicz, Carlos J. P. de Lucena
July 2001 **Crossroads**, Volume 7 Issue 4

Publisher: ACM Press

Full text available:  [html\(47.39 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

4

 **Session 7B: agent analysis and design: Extreme programming of multi-agent systems** 

Holger Knublauch

July 2002 **Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2**

Publisher: ACM Press

Full text available:  pdf(776.87 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The complexity of communication scenarios between agents make multi-agent systems difficult to build. Most of the existing Agent-Oriented Software Engineering methodologies face this complexity by guiding the developers through a rather waterfall-based process with a series of intermediate modeling artifacts. While these methodologies lead to executable prototypes relatively late and are expensive when requirements change, we explore a rather evolutionary approach with explicit support for chang ...

Keywords: agent-oriented software engineering

5 The joint ACM SIGPLAN student research competition & OOPSLA poster session: 

 **Dawn: must J2EE-Webapplications be untestable?**

Christian Dedeck, Andreas Spall, Sabine Winkler

November 2002 **Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**

Publisher: ACM Press

Full text available:  pdf(188.67 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

We present a cookbook for the development of J2EE-Web applications. This cookbook aims at the minimization of quality costs. It consists of procedures and guidelines in combination with a toolset. The classical software lifecycle presents *test* as a single phase in the development process. The cookbook presents *testing* as the motor of every other activity in development.

Keywords: J2EE, Web engineering, agile processes, open source tools, test-methodology

6 Full papers: Explicit programming 

 Avi Bryant, Andrew Catton, Kris De Volder, Gail C. Murphy

April 2002 **Proceedings of the 1st international conference on Aspect-oriented software development**

Publisher: ACM Press

Full text available:  pdf(856.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many design concepts can be expressed only indirectly in source code. When this occurs, a single concept at design results in a verbose amount of code that is scattered across the system structure. In this paper, we present explicit programming, an approach that enables a developer to introduce new vocabulary into the source to capture a design concept explicitly. An introduced vocabulary item modularizes the implementation details associated with a design concept, reducing the scattering of cod ...

7 CartaBlanca— a pure-Java, component-based systems simulation tool for coupled 

 **non-linear physics on unstructured grids**

W. B. VanderHeyden, E. D. Dendy, N. T. Padial-Collins

June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**

Publisher: ACM Press

Full text available: [pdf\(924.14 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes a component-based non-linear physical system simulation prototyping package written entirely in Java using object-oriented design to provide scientists and engineers a "developer-friendly" software environment for large-scale computational method and physical model development. The software design centers on the Jacobian-Free Newton-Krylov solution method surrounding a finite-volume treatment of conservation equations. This enables a clean component-based impl ...

Keywords: Jacobian, Java object oriented, Krylov, Newton, components, parallel, physics, solver, threads

8 Is ISSTA research relevant to industrial users? panel - ISSTA 2002: empowering the developer to be a tester too! 

Vincent Encontre

July 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis ISSTA '02**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: [pdf\(144.54 KB\)](#) Additional Information: [full citation](#), [abstract](#)

In this paper - scoped for the panel discussion at ISSTA 2002 - we are discussing some techniques to ease the adoption of testing techniques by the developers, by extending the debugging activity. We also briefly discuss a longer term vision where the same paradigm applies but at model level, when coding will be achieved using visual notations such as UML.

Keywords: JUnit, UML, debugging, runtime analysis, testing

9 Technical correspondence: Embedding built-in tests in hot spots of an object-oriented framework 

Taewoong Jeon, Hyon Woo Seung, Sungyoung Lee
August 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 8

Publisher: ACM Press

Full text available: [pdf\(793.62 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

This paper describes a scheme of encapsulating test support code as built-in test (BIT) components and embedding them into the hot spots of an object-oriented framework so that defects caused by the modification and extension of the framework can be detected effectively and efficiently through testing. The test components embedded into a framework in this way increase the testability of the framework by making it easy to control and observe the process of framework testing. The proposed technique ...

Keywords: built-in test (BIT), hook classes, object-oriented framework, testability

10 DrJava: a lightweight pedagogic environment for Java 

Eric Allen, Robert Cartwright, Brian Stoler

February 2002 **ACM SIGCSE Bulletin , Proceedings of the 33rd SIGCSE technical symposium on Computer science education SIGCSE '02**, Volume 34 Issue 1

Publisher: ACM Press

Full text available: [pdf\(451.91 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

DrJava is a pedagogic programming environment for Java that enables students to focus

on designing programs, rather than learning how to use the environment. The environment provides a simple interface based on a "read-eval-print loop" that enables a programmer to develop, test, and debug Java programs in an interactive, incremental fashion. This paper gives an overview of DrJava including its pedagogic rationale, functionality, and implementation.

11 Session: Making RUP agile



Michael Hirsch

November 2002 **OOPSLA 2002 Practitioners Reports**

Publisher: ACM Press

Full text available: [pdf\(926.79 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Unified Development Process (USDP) and especially its implementation by Rational Software Corporation, the Rational Unified Process (RUP), is a comprehensive process covering almost all aspects of software development projects. However, due to the great level of detail provided by RUP, many professionals do not consider RUP practical for small, fast paced projects. This paper reports the experiences with RUP on two small projects with teams of 3 to 5 developers. RUP proved to be adaptable to ...

12 Session: Agile techniques to avoid firefighting at a start-up



Joseph A. Blotner

November 2002 **OOPSLA 2002 Practitioners Reports**

Publisher: ACM Press

Full text available: [pdf\(187.60 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper documents the creation and evolution of the (modified) agile methodology implemented at Sabrix, Inc., a start-up enterprise-class software company. The company was delivering alpha quality software to early adopter customers on a weekly basis, using urgency as the primary driver for product development. New features and bug fixes were completed in a manner similar to a soccer game between two teams of five-year-olds, with everyone on the team chasing down the latest emergency. This pa ...

Keywords: SDLC, agile methodology, development process, start-up company

13 Testing monadic code with QuickCheck



Koen Claessen, John Hughes

October 2002 **Proceedings of the 2002 ACM SIGPLAN workshop on Haskell Haskell '02**

Publisher: ACM Press

Full text available: [pdf\(136.14 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

QuickCheck is a previously published random testing tool for Haskell programs. In this paper we show how to use it for testing monadic code, and in particular imperative code written using the *ST* monad. QuickCheck tests a program against a specification: we show that QuickCheck's specification language is sufficiently powerful to represent common forms of specifications: algebraic, model-based (both functional and relational), and pre-/post-conditional. Moreover, all these forms of specif ...

14 PLI workshops: Testing monadic code with QuickCheck



Koen Claessen, John Hughes

December 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 12

Publisher: ACM Press

Full text available: [pdf\(206.81 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

QuickCheck is a previously published random testing tool for Haskell programs. In this paper we show how to use it for testing monadic code, and in particular imperative code

written using the *ST* monad. QuickCheck tests a program against a specification: we show that QuickCheck's specification language is sufficiently powerful to represent common forms of specifications: algebraic, model-based (both functional and relational), and pre-/post-conditional. Moreover, all these forms of specif ...

15 Software and document engineering: The relevance of software documentation, tools and technologies: a survey



Andrew Forward, Timothy C. Lethbridge

November 2002 **Proceedings of the 2002 ACM symposium on Document engineering**

Publisher: ACM Press

Full text available: [pdf\(222.00 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper highlights the results of a survey of software professionals. One of the goals of this survey was to uncover the perceived relevance (or lack thereof) of software documentation, and the tools and technologies used to maintain, verify and validate such documents. The survey results highlight the preferences for and aversions against software documentation tools. Participants agree that documentation tools should seek to better extract knowledge from core resources. These resources incl ...

Keywords: documentation relevance, documentation survey, documentation technologies, program comprehension, software documentation, software engineering, software maintenance

16 Columns: Surfing the net for software engineering notes



Mark Doernhoefer

September 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.67 MB\)](#)

Additional Information: [full citation](#)

17 Short papers: Aspect-oriented programming with model checking



Naoyasu Ubayashi, Tetsuo Tamai

April 2002 **Proceedings of the 1st international conference on Aspect-oriented software development**

Publisher: ACM Press

Full text available: [pdf\(606.77 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Aspect-oriented programming (AOP) is a programming paradigm such that crosscutting concerns including synchronization policies, resource sharing and performance optimizations over objects are modularized as aspects that are separated from objects. A compiler, called a weaver, weaves aspects and objects together into a program. In AOP, however, it is not easy to verify the correctness of a woven program because crucial behaviors are strongly influenced by aspect descriptions. In order to deal wit ...

Keywords: checking frameworks, model checking, validation

Results 1 - 17 of 17

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [!\[\]\(e462f608b89f421f9d905728e26f6429_img.jpg\) Adobe Acrobat](#) [!\[\]\(d23a1e673062c7f9dd7f31b0bf3952ee_img.jpg\) QuickTime](#) [!\[\]\(b6decc5949267b59236e06e16f30320f_img.jpg\) Windows Media Player](#) [!\[\]\(1302b7c44ebac71a8fe8000e9986c651_img.jpg\) Real Player](#)

acm PORTAL
USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: The ACM Digital Library The Guide

Unit testing

http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=809968&CFTOKEN=683...

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [Unit testing](#)

Found 66,940 of 185,178

Sort results by

relevance

Save results to a Binder

Try an [Advanced Search](#)

Display results

expanded form

Search Tips
 Open results in a new window

Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: **1** [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

1 [Validation and verification: Unit testing in multi-agent systems using mock agents and aspects](#)



Roberta Coelho, Uirá Kulesza, Arndt von Staa, Carlos Lucena
May 2006 **Proceedings of the 2006 international workshop on Software engineering for large-scale multi-agent systems SELMAS '06**

Publisher: ACM Press

Full text available: [pdf\(169.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we present a unit testing approach for MASs based on the use of Mock Agents. Each Mock Agent is responsible for testing a single role of an agent under successful and exceptional scenarios. Aspect-oriented techniques are used, in our testing approach, to monitor and control the execution of asynchronous test cases. We present an implementation of our approach on top of JADE platform, and show how we extended JUnit test framework in order to execute JADE test cases.

Keywords: aspect oriented programming, dependability, mock objects, unit testing

2 [Unit testing: test early, test often](#)

Michael Olan

December 2003 **Journal of Computing Sciences in Colleges**, Volume 19 Issue 2

Publisher: Consortium for Computing Sciences in Colleges

Full text available: [pdf\(154.89 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Testing is a critical part of good software development, but often gets only minimal coverage in introductory programming courses. Unit testing and selected aspects of test-driven development can be used to improve learning and encourage emphasis on quality and correctness. Tools like JUnit significantly simplify the generation of test cases. An additional benefit for instructors is that these tools can also be used to automate project grading.

3 [Parameterized unit tests with unit meister](#)



Nikolai Tillmann, Wolfram Schulte

September 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13**, Volume 30 Issue 5

Publisher: ACM Press

Full text available:  pdf(50.24 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Parameterized unit tests extend the current industry practice of using closed unit tests defined as parameterless methods. Traditional closed unit tests are re-obtained by instantiating the parameterized unit tests. We have developed the prototype tool *Unit Meister*, which uses symbolic execution and constraint solving to automatically compute a minimal set of inputs that exercise a parameterized unit test given certain coverage criteria. In addition, the parameterized unit tests can be us ...

Keywords: algebraic data types, automatic test input generation, constraint solving, symbolic execution, unit testing

4 Effective unit testing

 Tim Burns
January 2001 **Ubiquity**, Volume 1 Issue 42

Publisher: ACM Press

Full text available:  html(6.50 KB) Additional Information: [full citation](#), [index terms](#)

5 Unit testing frameworks

 Robert E. Noonan, Richard H. Prosl
February 2002 **ACM SIGCSE Bulletin , Proceedings of the 33rd SIGCSE technical symposium on Computer science education SIGCSE '02**, Volume 34 Issue 1

Publisher: ACM Press

Full text available:  pdf(406.45 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

In this paper we explore frameworks for performing unit testing in Java. The vehicle for this exploration is a student-written, skeleton program developed for the *Computer Graphics* course. Our analysis of this one experiment leads us to speculate what benefits in program development and design might accrue by requiring students to unit test their own programs.

6 Research papers: test & analysis I: An intensional approach to the specification of test cases for database applications

 David Willmor, Suzanne M. Embury
May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06**

Publisher: ACM Press

Full text available:  pdf(360.22 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

When testing database applications, in addition to creating in-memory fixtures it is also necessary to create an initial database state that is appropriate for each test case. Current approaches either require exact database states to be specified in advance, or else generate a single initial state (under guidance from the user) that is intended to be suitable for execution of all test cases. The first method allows large test suites to be executed in batch, but requires considerable programmer ...

Keywords: database testing, databases, software testing

7 Economics of test automation: test cost and effectiveness: Advanced unit testing:

 how to scale up a unit test framework

Cyrille Artho, Armin Biere

May 2006 **Proceedings of the 2006 international workshop on Automation of software test AST '06**

Publisher: ACM Press

Full text available: [pdf\(200.19 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Unit testing is a scalable and effective way to uncover software faults. In the JNuke project, automated regression tests combined with coverage measurement ensured high code quality throughout the project. By using a custom testing environment, functionality was extended beyond what is commonly available by unit test frameworks. Low-overhead memory leak detection was implemented through wrapping. Automated support for log files made it possible to track the internal state of objects, which is o ...

Keywords: system testing, test framework, unit testing

8 Reviewed papers: An experience in integrating automated unit testing practices in an 

 introductory programming course

Elena García Barriocanal, Miguel-Ángel Sicilia Urbán, Ignacio Aedo Cuevas, Paloma Díaz Pérez

December 2002 **ACM SIGCSE Bulletin**, Volume 34 Issue 4

Publisher: ACM Press

Full text available: [pdf\(63.49 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Unit testing is one of the core practices in the Extreme Programming lightweight software development method, and it is usually carried out with the help of software frameworks that ease the construction of test cases as an integral part of programming tasks. This work describes our first results in studying the integration of automated unit testing practices in conventional 'introduction to programming' laboratories. Since the work used a classical procedural language in the course's assignment ...

9 TestGen — testing tool for Ada Designs and Ada code 

 Thomas S. Radi

July 1989 **Proceedings of the sixth Washington Ada symposium on Ada**

Publisher: ACM Press

Full text available: [pdf\(611.63 KB\)](#) Additional Information: [full citation](#), [index terms](#)

10 Parameterized unit tests 

 Nikolai Tillmann, Wolfram Schulte

September 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13**, Volume 30 Issue 5

Publisher: ACM Press

Full text available: [pdf\(131.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Parameterized unit tests extend the current industry practice of using closed unit tests defined as parameterless methods. Parameterized unit tests separate two concerns: 1) They specify the external behavior of the involved methods for all test arguments. 2) Test cases can be re-obtained as traditional closed unit tests by instantiating the parameterized unit tests. Symbolic execution and constraint solving can be used to automatically choose a minimal set of inputs that exercise a parameterize ...

Keywords: algebraic data types, automatic test input generation, constraint solving, symbolic execution, unit testing

11 Towards a BPEL unit testing framework

 Philip Mayer, Daniel Lübke
July 2006 **Proceedings of the 2006 workshop on Testing, analysis, and verification of web services and applications TAV-WEB '06**

Publisher: ACM Press

Full text available:  pdf(198.95 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Business Process Execution Language (BPEL) is emerging as the new standard in Web service composition. As more and more workflows are modelled using BPEL, unit-testing these compositions becomes increasingly important. However, little research has been done in this area and no frameworks comparable to the xUnit family are available. In this paper, we propose a layer-based approach to creating frameworks for repeatable, white-box BPEL unit testing, which we use for the development of a new te ...

Keywords: BPEL, BPELUnit, composition, orchestration, testing, unit testing

12 Incorporating varying test costs and fault severities into test case prioritization

Sebastian Elbaum, Alexey Malishevsky, Gregg Rothermel

July 2001 **Proceedings of the 23rd International Conference on Software Engineering**

Publisher: IEEE Computer Society

Full text available:  pdf(885.24 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)

Test case prioritization techniques schedule test cases for regression testing in an order that increases their ability to meet some performance goal. One performance goal, *rate of fault detection*, measures how quickly faults are detected within the testing process. In previous work we provided a metric, APFD, for measuring rate of fault detection, and techniques for prioritizing test cases to improve APFD, and reported the results of experiments using those techniques. This met ...

Keywords: fault severity, rate of fault detection, regression testing, test case prioritization, test cost

13 Session 6: test automation: Automated testing of stochastic systems: a statistically grounded approach

Hana Ševčíková, Alan Borning, David Socha, Wolf-Gideon Bleek

July 2006 **Proceedings of the 2006 international symposium on Software testing and analysis ISSTA'06**

Publisher: ACM Press

Full text available:  pdf(215.70 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Automated tests can play a key role in ensuring system quality in software development. However, significant problems arise in automating tests of stochastic algorithms. Normally, developers write tests that simply check whether the actual result is equal to the expected result (perhaps within some tolerance). But for stochastic algorithms, restricting ourselves in this way severely limits the kinds of tests we can write: either to trivial tests, or to fragile and hard-to-understand tests that re ...

Keywords: hypothesis testing, software engineering, software testing, stochastic algorithms, unit tests

Session 3: modular reasoning: Subdomain testing of units and systems with state**Dick Hamlet****July 2006 Proceedings of the 2006 international symposium on Software testing and analysis ISSTA'06****Publisher:** ACM PressFull text available:  [pdf\(397.12 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper extends basic software-testing theory to software components and adds explicit state to the theory. The resulting theory is simple enough to abstractly model the construction of systems from their parts ('units'). It provides an unconventional insight into the relationship between testing units and testing systems. Experiments exploring the theory support the following conclusions:

- Units should be independent, more like what are called "components" than subroutines or object-...

Keywords: persistent state, testing theory, unit/system testing**15 Session 1: Tool support for randomized unit testing****James H. Andrews, Susmita Haldar, Yong Lei, Felix Chun Hang Li****July 2006 Proceedings of the 1st international workshop on Random testing RT '06****Publisher:** ACM PressFull text available:  [pdf\(336.44 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

There are several problem areas that must be addressed when applying randomization to unit testing. As yet no general, fully automated solution that works for all units has been proposed. We therefore have developed RUTE-J, a Java package intended to help programmers do randomized unit testing in Java. In this paper, we describe RUTE-J and illustrate how it supports the development of per-unit solutions for the problems of randomized unit testing. We report on an experiment in which we applied R ...

Keywords: randomized testing, unit testing**16 Using testing and JUnit across the curriculum****Michael Wick, Daniel Stevenson, Paul Wagner****February 2005 ACM SIGCSE Bulletin , Proceedings of the 36th SIGCSE technical symposium on Computer science education SIGCSE '05, Volume 37 Issue 1****Publisher:** ACM PressFull text available:  [pdf\(219.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

While the usage of unit-testing frameworks such as JUnit has greatly increased over the last several years, it is not immediately apparent to students and instructors how to best use tools like JUnit and how to integrate testing across a computer science curriculum. We have worked over the last four semesters to infuse testing and JUnit across our curriculum, building from having students use JUnit to having them write their own test cases to building larger integration and use case testing syst ...

Keywords: JUnit, testing, unit testing, unit testing frameworks**17 A case for test-code generation in model-driven systems****Matthew J. Rutherford, Alexander L. Wolf****September 2003 Proceedings of the second international conference on Generative programming and component engineering GPCE '03****Publisher:** Springer-Verlag New York, Inc.

Full text available:  pdf(283.86 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A primary goal of generative programming and model-driven development is to raise the level of abstraction at which designers and developers interact with the software systems they are building. During initial development, the benefits of abstraction are clear. However, during testing and maintenance, increased distance from the implementation can be a disadvantage. We view test cases and test harnesses as an essential bridge between the high-level specifications and the implementation. As such, ...

18 Inter-class testing of O-O software in the presence of polymorphism 

Amit Paradkar

November 1996 **Proceedings of the 1996 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available:  pdf(206.67 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes an algorithm for testing polymorphic interactions between classes in object-oriented (**O-O**) software. We propose an Augmented Object Relationship Diagram (**AORD**), which is an extension of a previously reported **ORD** representation, to facilitate the algorithm. **AORD** is a graph representation of the class structure augmented with additional information about the methods defined in each class. This additional information consists of call-graph of all the method ...

19 Automated testing of virtual reality application interfaces 

 Allen Bierbaum, Patrick Hartling, Carolina Cruz-Neira

May 2003 **Proceedings of the workshop on Virtual environments 2003 EGVE '03**

Publisher: ACM Press

Full text available:  pdf(601.80 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe a technique for supporting testing of the interaction aspect of virtual reality (VR) applications. Testing is a fundamental development practice that forms the basis of many software engineering methodologies. It is used to ensure the correct behavior of applications. Currently, there is no common pattern for automated testing of VR application interaction. We review current software engineering practices used in testing and explore how they may be applied to the specific realm of VR ...

Keywords: VR Juggler, extreme programming, unit testing

20 Embedded hardware and software self-testing methodologies for processor cores 

 Li Chen, Sujit Dey, Pablo Sanchez, Krishna Sekar, Ying Cheng

June 2000 **Proceedings of the 37th conference on Design automation**

Publisher: ACM Press

Full text available:  pdf(103.87 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

At-speed testing of GHz processors using external testers may not be technically and economically feasible. Hence, there is an emerging need for low-cost, high-quality self-test methodologies, which can be used by processors to test themselves at-speed. Currently, Built-In Self-Test (BIST) is the primary self-test methodology available and is widely used for testing embedded memory cores. In this paper, we report our experiences in applying a commercial BIST methodology to two processor cor ...

Useful downloads: [!\[\]\(51423b03ed5dbe39f78a50141211e114_img.jpg\) Adobe Acrobat](#) [!\[\]\(689c07aaf2503d12f145ee48cf00904e_img.jpg\) QuickTime](#) [!\[\]\(0543d17b28a15d82503ad7e5332658ec_img.jpg\) Windows Media Player](#) [!\[\]\(2269ee29a8da0dab6e0fc0b2812b7c0f_img.jpg\) Real Player](#)

[Sign in](#)
[Web](#) [Images](#) [Video^{New!}](#) [News](#) [Maps](#) [more »](#)

junit

[Advanced Search](#)
[Preferences](#)
WebResults 1 - 10 of about 11,400,000 for **junit**. (0.11 seconds)**JUnit, Testing Resources for Extreme Programming**

This site is dedicated to software developers using **JUnit** or one of the other XUnit testing frameworks. They are tools for Extreme Programming advocates.

[www.junit.org/](#) - 15k - [Cached](#) - [Similar pages](#)

[Articles](#) - [www.junit.org/news/article/index.htm](#)

[Extensions](#) - [www.junit.org/news/extension/index.htm](#)

[Watij](#) - [Web Application Testing](#) ... -

[www.junit.org/.../web/index.htm](#)

[IDEs](#) - [www.junit.org/news/ide/index.htm](#)

[More results from www.junit.org](#) »

Sponsored Links**Software Test Automation**

Get the Latest Software Test Tools
 Download Free Trial Today
[www.RedstoneSoftware.com](#)

Improve Your Java Testing

Identify exactly what areas of your code have been tested. Demo/Papers
[www.mccabe.com](#)

Java Test Coverage Tools

Very low overhead, large systems
 Client, Server (J2EE) or embedded
[www.semanticdesigns.com](#)

Manage JUnit Statistics

Set up **JUnit** statistics in 5 min,
 get Parabuild server now
[www.viewtier.com](#)

JUnit

JUnit is a simple framework to write repeatable tests. ... To get started with unit testing and **JUnit** read the article: **JUnit** Cookbook. ...

[junit.sourceforge.net/](#) - 5k - [Cached](#) - [Similar pages](#)

JUnit Cookbook

JUnit tests do not require human judgment to interpret, and it is easy to run many of them at the same time. When you need to test something, here is what ...

[junit.sourceforge.net/doc/cookbook/cookbook.htm](#) - 6k -

[Cached](#) - [Similar pages](#)

[[More results from junit.sourceforge.net](#)]

JUnit Primer

A primer describing how to write and run **JUnit** tests.

[clarkware.com/articles/JUnitPrimer.html](#) - 27k - [Cached](#) - [Similar pages](#)

XP Software

JUnitPerf is a set of **Junit** decorators for automated load testing. ... **JSystem** is a framework for writing and running automated tests, base on **JUnit**. ...

[www.xprogramming.com/software.htm](#) - 71k - [Cached](#) - [Similar pages](#)

JUnit Task

Note: You must have **junit.jar** and the class files for the **<junit>** task in the same ... Put both **junit.jar** and the optional tasks jar file in **ANT_HOME/lib** ...

[ant.apache.org/manual/OptionalTasks/junit.html](#) - 21k - [Cached](#) - [Similar pages](#)

JUnit best practices

Extreme Programming's rise in popularity among the Java community has prompted more development teams to use **JUnit**: a simple test framework for building and ...

[www.javaworld.com/javaworld/jw-12-2000/jw-1221-junit.html](#) - [Similar pages](#)

SourceForge.net: JUnit

The world's largest development and download repository of Open Source code and applications.

[sourceforge.net/projects/junit/](#) - 27k - [Cached](#) - [Similar pages](#)

Test Infected:

Using **JUnit** you can cheaply and incrementally build a test suite that will help ... To write tests you need to get the latest copy **JUnit** (or write your own ...
members.pingnet.ch/gamma/junit.htm - 41k - Cached - Similar pages

JUnit - Wikipedia, the free encyclopedia

JUnit has spawned its own ecosystem of **JUnit** extensions. Experience gained with **JUnit** has been important in the development of test-driven development, ...
en.wikipedia.org/wiki/JUnit - 16k - Cached - Similar pages

Gooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 [**Next**](#)

Free! Speed up the web. [Download the Google Web Accelerator](#).

junit

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google

[Sign in](#)[Web](#) [Images](#) [Video^{New!}](#) [News](#) [Maps](#) [more »](#)

unit testing

[Advanced Search](#)
[Preferences](#)**Web**Results 1 - 10 of about 132,000,000 for **unit testing**. (0.11 seconds)**Unit Testing**

www.google.com/jobs Google engineering might be the place for you. Experts needed!

Sponsored Link

Sponsored Links

Unit test - Wikipedia, the free encyclopedia

The goal of **unit testing** is to isolate each part of the program and show that the ... **Unit testing** is the building block to **test** driven development (TDD). ...
en.wikipedia.org/wiki/Unit_testing - 27k - [Cached](#) - [Similar pages](#)

Automated Unit Testing

Automate java **unit testing**
 Test creation, execution, reporting
www.agitar.com

JUnit, Testing Resources for Extreme Programming

Welcome to JUnit.org. This site is dedicated to software developers using JUnit or one of the other XUnit **testing** frameworks...
www.junit.org/ - 15k - [Cached](#) - [Similar pages](#)

Improve Your Testing - IQ

Identify exactly what areas of your code have been tested. Demo/Paper
www.mccabe.com

Extreme Rules

First you should create or download a **unit test** framework to be able to create automated **unit** tests suites. Second you should **test** all classes in the system ...
www.extremeprogramming.org/rules/unittests.html - 8k - [Cached](#) - [Similar pages](#)

Unit Testing

Jtest Automates JUnit Test Creation & Execution; Checks 500+ Java Rules
www.parasoft.com

Chapter 13. Unit Testing

Chapter 13. **Unit Testing**. 13.1. Introduction to Roman numerals · 13.2. Diving in · 13.3. Introducing romantest.py · 13.4. **Testing** for success · 13.5. ...
www.diveintopython.org/unit_testing/index.html - 12k - [Cached](#) - [Similar pages](#)

Automate Unit Testing

TBrun automates **unit testing** for C/C++, C# & Ada 83/95.
www.ldra.com

Test Infected:

Using JUnit you can cheaply and incrementally build a **test** suite that will help you ... you get from writing and saving and rerunning your own **unit** tests. ...
junit.sourceforge.net/doc/testinfected/testing.htm - 32k - [Cached](#) - [Similar pages](#)

Functional Testing

Black box, **Unit**, Integration, Acceptance & Regression Testing
www.readytogo.com

Six Rules of Unit Testing

Last updated: 25/7/02; 6:07:05 PM. The Desktop Fishbowl Charles blogs all the random nerd stuff he can find. Six Rules of **Unit Testing** ...
radio.weblogs.com/0100190/stories/2002/07/25/sixRulesOfUnitTesting.html - 13k - [Cached](#) - [Similar pages](#)

C/C++ Unit Testing

Test Automation for embedded device software. No scripting required.
www.crosstest.com

DB Testing

Notes on **unit testing** database functionality. ... These are my notes on how I've gone about **unit testing** database functionality. The examples are in Java, ...
www.dallaway.com/acad/dbunit.html - 23k - [Cached](#) - [Similar pages](#)

Unit Testing

Looking to find **unit testing**? Visit our **unit testing** guide.
UnitTesting.info

[More Sponsored Links »](#)**Advanced Unit Test, Part V - Unit Test Patterns - The Code Project ...**

An Introduction To The Concept Of Unit Test Patterns.

www.codeproject.com/gen/design/autp5.asp - 84k - [Cached](#) - [Similar pages](#)

[develop:unit_testing_in_php \[phpPatterns\]](#)

Unit testing is one of the practical facets of eXtreme programming. ... In this article I'll look at a script which makes unit testing a pleasure rather ...

www.phppatterns.com/docs/develop/unit_testing_in_php - 32k - [Cached](#) - [Similar pages](#)

[NUnit - Home](#)

NUnit is a **unit-testing** framework for all .Net languages. Initially ported from JUnit, the current production release, version 2.2, is the fourth major ...

www.nunit.org/ - 5k - [Cached](#) - [Similar pages](#)

Gooooooooogle ►

Result Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google